



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/784,297	02/24/2004	Takashi Tameshige	NITT.0196	5051

7590 11/16/2007  
Stanley P. Fisher  
Reed Smith LLP  
Suite 1400  
3110 Fairview Park Drive  
Falls Church, VA 22042-4503

EXAMINER
----------

DICKER, DENNIS T

ART UNIT	PAPER NUMBER
----------	--------------

2625

MAIL DATE	DELIVERY MODE
-----------	---------------

11/16/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	Application No. 10/784,297	Applicant(s) TAMESHIGE ET AL.	
	Examiner Dennis Dicker	Art Unit 2625	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 24 February 2004.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-10 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-10 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 24 February 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All    b) ☐ Some \*    c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date <u>2/24/2004; 8/2/2007</u> . | 6) <input type="checkbox"/> Other: _____  |

## DETAILED ACTION

### *Claim Rejections - 35 USC § 103*

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-3, 5-8 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kataoka (hereinafter "Kataoka '462" 2001/0056462) in view of Yamamoto Soichi (hereinafter "Soichi '203" JP 09-097203 [Translation]).

With respect to **Claim 1**, Kataoka '462 teaches a method for executing a job loaded into a client machine (**i.e., 150 of Fig. 2**) on a server machine (**i.e., 110 of Fig. 2**) that is different in a computer environment from the client machine (**i.e., Para 0008-0009, method of executing an application on a computer from a computer with a different environment**), said method comprising the steps of: allowing the client machine to issue to the server machine a job execution request for executing the job (**i.e., Para 0012, Client machine makes an execution request for server machine**), the job execution request being accompanied by environment information (**i.e., Para 0083, application program logic and local environment identifying logic are loaded simultaneously**) and job execution statements for the job to be executed (**i.e., Para 0013, starting command for the application program**).

Kataoka '462 does not explicitly teach a method where the environment information including a volume logical path and a volume physical path and allowing the server machine to control to assign a server side volume corresponding to a client side and transfer input data on the client side volume to the server side, to convert the environment information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine, and to execute the job.

However, the mentioned claimed limitations are well known in the art as evidenced by Soichi '203. In particular, Soichi '203 teaches the use of the environment information including a volume logical path and a volume physical path (**i.e., Para 0016, File having a logical and physical path**) and allowing the server machine to control to assign a server side volume corresponding to a client side and transfer input data on the client side volume to the server side (**i.e., Para 0019, Job management server directs the file name to the physical device name**), to convert the environment information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine, and to execute the job (**i.e., Para 0019-0020, changes the logical and physical path of the job are changed to the server and activating the job**)

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the method of Kataoka '462 as taught by Soichi '203 since Soichi '203 suggested in Para 0015 that such a modification would

Art Unit: 2625

allow a distributing processing system to obtain the processing result more than a large sized computer.

As pertaining to **Claim 2**, Kataoka '462 teaches a method for causing a client machine to issue a job execution request in a system (i.e., **Para 0008-009, Method of executing an application on a computer from a computer with a different environment**), the system comprising the client machine (i.e., **150 of Fig. 2**) into which a job is loaded and a server machine (i.e., **110 of Fig. 2**) that is different in a computer environment from the client machine, said method comprising the steps of:

Allowing the client machine to determine according to policy information (i.e., **Para 00and 0048, 0051 and Fig. 3, Client machine 110 has historical information**) whether the job is to be executed on the server machine allowing the client machine to issue a job execution request for the job to the server machine if it is determined that the job is to be executed on the server machine (i.e., **0061-0065, Historical information on 110 indicates the version of the software and the command is executed which determines if the program is to be ran or not** ), the job execution request being accompanied by environment information and job execution statements for the job to be executed information (i.e., **Para 0083,application program logic and local environment identifying logic are loaded simultaneously**)

Kataoka '462 does not explicitly teach a method where the environment information including a volume logical path and a volume physical path, allowing the client machine to transfer input data to a server side volume corresponding to a client side volume, and allowing the client machine to receive the execution result of the job

Art Unit: 2625

that is executed according to the environment information and the job execution statements for which the volume logical path and the volume physical path are replaced by the corresponding information for the server machine, and the resulting billing information for the execution result.

However, the mentioned claimed limitations are well known in the art as evidenced by Soichi '203. In particular, Soichi '203 teaches a method where the environment information including a volume logical path and a volume physical path **(i.e., Para 0016, activation means includes a file having a logical and physical path)**, allowing the client machine to transfer input data to a server side volume corresponding to a client side volume **(i.e., Para 0019, Job management server directs the file name to the physical device name)**, and allowing the client machine to receive the execution result of the job that is executed **(i.e., Para 0020 , job report of activation result)** according to the environment information and the job execution statements for which the volume logical path and the volume physical path are replaced by the corresponding information for the server machine **(i.e., ,Para 0020 and 0028, the logical and physical resource translation table changes the name to the physical device name)**, and the resulting billing information for the execution result **(i.e., Para 0016 , Resource translation table changes the I/O file to the physical device).**

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the method of Kataoka '462 as taught by Soichi '203 since Soichi '203 suggested in Para 0015 that such a modification would

Art Unit: 2625

allow a distributing processing system to obtain the processing result more than a large sized computer.

With respect to Claim 3, Kataoka '462 teaches a method for causing a server machine to execute a job in a system (i.e., **Para 0007, computer in a system executing a program**), the system comprising a client machine into which the job is loaded and the server machine that is different in a computer environment from the client machine (i.e., **Para 0008-009, Method of executing an application on a computer from a client to a server**), said method comprising the steps of allowing the server machine to receive a job execution request for the job from the client machine (i.e., **0061-0065, Historical information on 110 indicates the version of the software and the command is executed which determines if the program is to be ran or not**), the job execution request being accompanied by environment information and job execution statements for the job to be executed (i.e., **Para 0083, application program logic and local environment identifying logic are loaded simultaneously**).

Kataoka '462 does not explicitly teach a method where the environment information including a volume logical path and a volume physical path; to control to assign a server side volume corresponding to a client side volume and transfer input data on the client side volume to the server side volume; to convert the environment information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine; and to execute the job.

However, the mentioned claimed limitations are well known in the art as evidenced by Soichi '203. In particular, Soichi '203 teaches the use of a method where the environment information including a volume logical path and a volume physical path **(i.e., Para 0016, activation means includes a file having a logical and physical path)**; to control to assign a server side volume corresponding to a client side volume and transfer input data on the client side volume to the server side volume **(i.e., Para 0019, Job management server directs the file name to the physical device name)**; to convert the environment information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine; and to execute the job. **(i.e., Para 0025, the logical and physical resource translation table changes the name to the physical device name and then the job is performed).**

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the method of Kataoka '462 as taught by Soichi '203 since Soichi '203 suggested in Para 0015 that such a modification would allow a distributing processing system to obtain the processing result more than a large sized computer.

With respect to **Claim 5**, Kataoka '462 teaches a method wherein the environment information includes the names of programs executed for the job and the information about the versions of the programs **(i.e., Para 0053 and 233 and 239 of Fig. 4, Environment Information includes Name of software and version)**, and wherein the server machine determines whether the versions of the programs are



Art Unit: 2625

installed on the server machine (i.e., **Para 0065-0068, code for checking a particular version of software**) and installs any uninstalled program on the server machine (i.e., **Para 0090-0092, software BAP 1.0 is downloaded to a local directory and installed**).

With respect to **Claim 6**, Kataoka '462 does not explicitly teach a method, wherein the server machine executes the job in accordance with the job execution statements for which the amount of computer resource use described in the job execution statement is changed in compliance with the information about a service level agreement.

However, the mentioned claimed limitations are well known in the art as evidenced by Soichi '203. In particular, Soichi '203 teaches the use of a server machine which executes the job in accordance with the job execution statements (i.e., **Para 0019, job activation sever to which activation of a job was directed in accordance with a job activation means**) for which the amount of computer resource use described in the job execution statement is changed in compliance with the information about a service level agreement (i.e., **Para 0019 , job activation means changes file name described by the program with the resource translation table**).

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the method of Kataoka '462 as taught by Soichi '203 since Soichi '203 suggested in Para 0015 that such a modification would allow a distributing processing system to obtain the processing result more than a large sized computer.

Art Unit: 2625

With respect to **Claim 7**, Kataoka '462 teaches a program that runs on a client machine in a system (i.e., **Para 0038-0039, ROM 14 of Fig. 1 which comprises the program of the invention is can be implemented in a client machine**) comprising the client machine (i.e., **150 of Fig. 2**) into which a job is loaded and a server machine (i.e., **110 of Fig. 2**) that is different in a computer environment from the client machine (i.e., **Para 0008-009, Method of executing an application on a computer from a computer with a different environment**), said program including: a function for determining whether the server machine is to execute the job in compliance with policy information (i.e., **Para 0012, Client machine makes an execution request for server machine**), a function for issuing a job execution request for the job to the server machine if it is determined that the server machine is to execute the job (i.e., **Para 0013, starting command for the application program**), wherein the job execution request is accompanied by environment information and job execution statements for the job to be executed (i.e., **Para 0083,application program logic and local environment identifying logic are loaded simultaneously**).

Kataoka '462 does not explicitly teach a program where the environment information includes a volume logical path and a volume physical path; a function for transferring input data to a server side volume corresponding to a client side volume; and a function for receiving the execution result of the job that is executed according to the environment information and the job execution statements for which the volume logical path and the volume physical path are replaced by the corresponding information for the server machine, and the resulting billing information for the execution result.

Art Unit: 2625

However, the mentioned claimed limitations are well known in the art as evidenced by Soichi '203. In particular, Soichi '203 teaches the use of a program where the environment information includes a volume logical path and a volume physical path (i.e., **Para 0016, File having a logical and physical path**); a function for transferring input data to a server side volume corresponding to a client side volume (i.e., **Para 0019, Job management server directs the file name to the physical device name**; and a function for receiving the execution result of the job (i.e., **Para 0020 , job report of activation result**) that is executed according to the environment information and the job execution statements for which the volume logical path and the volume physical path are replaced by the corresponding information for the server machine i.e., **Para 0020 and 0028, the logical and physical resource translation table changes the name to the physical device name**), and the resulting billing information for the execution result (i.e., **Para 0016 , Resource translation table changes the I/O file to the physical device**).

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the method of Kataoka '462 as taught by Soichi '203 since Soichi '203 suggested in Para 0015 that such a modification would allow a distributing processing system to obtain the processing result more than a large sized computer.

With respect to Claim 8, Kataoka '462 teaches a program that runs on a server machine in a system (i.e., **Para 0038-0039, ROM 14 of Fig. 1 which comprises the program of the invention is can be implemented in a server machine in a system**)

Art Unit: 2625

comprising a client machine (i.e., **150 of Fig. 2**) into which a job is loaded and the server machine (i.e., **110 of Fig. 2**) that is different in a computer environment from the client machine (i.e., **Para 0008-009, Method of executing an application on a computer from a computer with a different environment**), said program including: a function for receiving a job execution request for the job from the client machine (i.e., **Para 0012, Client machine makes an execution request for server machine**), wherein the job execution request is accompanied by environment information and job execution statements for the job to be executed, (i.e., **Para 0083,application program logic and local environment identifying logic are loaded simultaneously**).

Kataoka '462 does not explicitly teach a program wherein the environment information includes a volume logical path and a volume physical path; a function for controlling to assign a server side volume corresponding to a client side volume and transfer input data on the client side volume to the server side volume; a function for converting the environment information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine; and a function for executing the job.

However, the mentioned claimed limitations are well known in the art as evidenced by Soichi '203. In particular, Soichi '203 teaches the use of a program where the environment information includes a volume logical path and a volume physical path (i.e., **Para 0016, activation means includes a file having a logical and physical path**); a function for controlling to assign a server side volume corresponding to a client

Art Unit: 2625

side volume and transfer input data on the client side volume to the server side volume **(i.e., Para 0019, Job management server directs the file name to the physical device name)**; a function for converting the environment information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine; and a function for executing the job **(i.e., ,Para 0025, the logical and physical resource translation table changes the name to the physical device name and then the job is performed).**

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the method of Kataoka '462 as taught by Soichi '203 since Soichi '203 suggested in Para 0015 that such a modification would allow a distributing processing system to obtain the processing result more than a large sized computer.

With respect to Claim 10, Kataoka '462 teaches a program wherein the environment information includes the names of programs executed for the job and the information about the versions of the programs **(i.e., Para 0053 and 233 and 239 of Fig. 4, Environment Information includes Name of software and version)**, and wherein the server machine causes the program to realize a function for determining whether the versions of the programs are installed on the server machine **(i.e., Para 0065-0068, code for checking a particular version of software)**; and a function for installing any uninstalled program on the server machine **(i.e., Para 0090-0092, software BAP 1.0 is downloaded to a local directory and installed).**

2. Claims 4 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kataoka '462 in view of Soichi '203 and further in view of Loomans (Hereinafter "Loomans '605" 6,393,605).

With respect to **Claim 4**, Kataoka '462 teaches a method for causing a server machine to execute a job in a system (i.e., **Para 0007, Computer system causing a computer to execute a program**), the system comprising a client machine into which the job is loaded and the server machine that is different in a computer environment from the client machine (i.e., **Para 0008-009, Method of executing an application on a computer from a client to a server**), said method comprising the steps of allowing the server machine to receive a job execution request for the job from the client machine (i.e., **0061-0065, Historical information on 110 indicates the version of the software and the command is executed which determines if the program is to be ran or not**), the job execution request being accompanied by environment information and job execution statements for the job to be executed (i.e., **Para 0083,application program logic and local environment identifying logic are loaded simultaneously**)

Soichi '203 teaches the use of a method where the environment information including a volume logical path and a volume physical path (i.e., **Para 0016, activation means includes a file having a logical and physical path**); to control to assign a server side volume corresponding to a client side volume and transfer input data on the client side volume to the server side volume (i.e., **Para 0019, Job management server directs the file name to the physical device name**); to convert the environment

Art Unit: 2625

information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine; and to execute the job. **(i.e., Para 0025, the logical and physical resource translation table changes the name to the physical device name and then the job is performed)**

Kataoka '462 or Soichi '203 do not teach a method wherein the server machine comprises a plurality of logically partitioned logical computers, and wherein, when the job execution request is received, the job is executed on a logical computer that can interpret and execute the job execution statements.

However, the mentioned claimed limitations are well known in the art as evidenced by Loomans '605. In particular, Loomans '605 teaches the use of server machine which comprises a plurality of logically partitioned logical computers **(i.e., 202 of Fig. 2, Computer comprising a plurality of UI elements)**, and wherein, when the job execution request is received, the job is executed on a logical computer that can interpret and execute the job execution statements **(i.e., Col 7 Lines 40-50, a UI element is selected by the application engine when ready to begin processing)**.

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the method of Kataoka '462 and Soichi '203 as taught by Loomans '605 since Loomans '605 suggested in Col 1 Line 25-29 that such a modification to the method would provide a highly interactive, robust, and scalable on demand network application.

Art Unit: 2625

With respect to **Claim 9**, Kataoka '462 teaches a program that runs on a server machine in a system (i.e., **Para 0038-0039, ROM 14 of Fig. 1 which comprises the program of the invention is can be implemented in a server machine in a system**) comprising a client machine (i.e., **150 of Fig. 2** into which a job is loaded and the server machine (i.e., **110 of Fig. 2**) that is different in a computer environment from the client machine (i.e., **Para 0008-009, Method of executing an application on a computer from a computer with a different environment**), said program including: a function for receiving a job execution request for the job from the client machine (i.e., **Para 0012, Client machine makes an execution request for server machine**), wherein the job execution request is accompanied by environment information and job execution statements for the job to be executed, (i.e., **Para 0083,application program logic and local environment identifying logic are loaded simultaneously**).

Soichi '203 teaches the use of a program where the environment information includes a volume logical path and a volume physical path (i.e., **Para 0016, activation means includes a file having a logical and physical path**); a function for controlling to assign a server side volume corresponding to a client side volume and transfer input data on the client side volume to the server side volume (i.e., **Para 0019, Job management server directs the file name to the physical device name**); a function for converting the environment information and the job execution statements so as to replace the information about the volume logical path and the volume physical path by the corresponding information for the server machine; and a function for executing the



job (i.e., **Para 0025, the logical and physical resource translation table changes the name to the physical device name and then the job is performed**).

Kataoka '462 or Soichi '203 do not teach a program wherein the server machine comprises a plurality of logically partitioned logical computers, and wherein, when the job execution request is received, the server machine causes the program to realize a function for executing the job on a logical computer that can interpret and execute the job execution statements.

However, the mentioned claimed limitations are well known in the art as evidenced by Loomans '605. In particular, Loomans '605 teaches a program (i.e., **Col. 3 Lines 61-Col 4 Lines 7, Program ran on client which loads user interface elements**) wherein the server machine comprises a plurality of logically partitioned logical computers (i.e., **202 of Fig. 2, Computer comprising a plurality of user interface elements**), and wherein, when the job execution request is received, the server machine causes the program to realize a function for executing the job on a logical computer that can interpret and execute the job execution statements (i.e., **Col 7 Lines 40-50, a UI element is selected by the application engine when ready to begin processing**).

In view of this, it would have been obvious to one having ordinary skill in the art at the time of invention was made to modify the program of Kataoka '462 and Soichi '203 as taught by Loomans '605 since Loomans '605 suggested in Col 1 Line 25-29 that such a modification to the method would provide a highly interactive, robust, and scalable on demand network application.

Art Unit: 2625

**Conclusion**

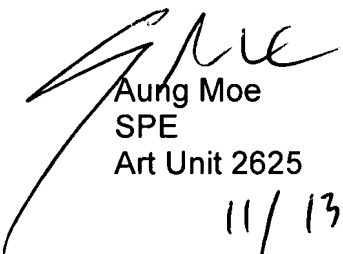
3. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis Dicker whose telephone number is (571) 270-3140. The examiner can normally be reached on Monday -Friday 7:30 A.M. to 5:00 P.M..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Aung Moe can be reached on (571) 272-7314. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DD  
November 13, 2007

  
Aung Moe  
SPE  
Art Unit 2625  
11/13/07